

Research Article

Approximately Nearest Neighborhood Image Search Using Unsupervised Hashing via Homogeneous Kernels

Jun-Yi Li  and Jian-Hua Li

Shanghai Jiaotong University Electrical and Electronic Engineering College, Shanghai, China

Correspondence should be addressed to Jun-Yi Li; leejy2006@163.com

Received 5 April 2017; Revised 28 May 2017; Accepted 29 April 2018; Published 23 May 2018

Academic Editor: Dimitri Volchenkov

Copyright © 2018 Jun-Yi Li and Jian-Hua Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose an approximation search algorithm that uses additive homogeneous kernel mapping to search for an image approximation based on kernelized locality-sensitive hashing. To address problems related to the unstable search accuracy of an unsupervised image hashing function and degradation of the search-time performance with increases in the number of hashing bits, we propose a method that combines additive explicit homogeneous kernel mapping and image feature histograms to construct a search algorithm based on a locality-sensitive hashing function. Moreover, to address the problem of semantic gaps caused by using image data that lack type information in semantic modeling, we describe an approximation searching algorithm based on the homogeneous kernel mapping of similarities between pairs of images and dissimilar constraint relationships. Our image search experiments confirmed that the proposed algorithm can construct a locality-sensitive hash function more accurately, thereby effectively improving the similarity search performance.

1. Introduction

At present, the availability of visual data from the Internet is increasing rapidly, including scientific images, photo galleries from online communities, and news photo galleries. Thus, rapid content-based searches for images and videos in large databases are necessary. Nearest neighbor search is a prerequisite for image search, where the aim is to find examples with the highest similarity and to return the query result. In fact, query items can be obtained by brute force search based on the nearest neighbors in large databases, which is followed by similarity-based classification. However, brute force search will incur considerable costs if there are a large number of items or when the computational load of the similarity function is very large. Thus, in visual applications, the most effective method involves representing data in a structured form or mapping the data onto a high-dimensional space. However, in large-scale image search tasks, unsatisfactory results may be obtained in the high-dimensional space where the data structures are used for exact search. In addition to the conventional methods used for learning a distance metric to approximately guarantee linear time consumption, we need

to develop appropriate distance metric methods based on specific constraint conditions. To address the problems of image data representation and distance metric learning methods, we must achieve a good balance between the applicability of an algorithm and its computational complexity. Thus, we propose a universal algorithm framework where nearest neighbors are searched rapidly using a homogeneous kernel map and metric learning.

Similarity search techniques have been developed to facilitate large-scale image searches, although sometimes at the expense of the accuracy of the predictions obtained [1–8], where the locality-sensitive hashing (LSH) algorithm is a typical example [9, 10]. In the LSH function, the query items must be within $(1 + \epsilon)$ times the distance from the actual similar points, such as formula $d(q, o') \leq (1 + \epsilon)d(q, o)$, and the number of queries has a linear relationship with the total number n . The LSH function ensures a high probability of collision between similar examples. Several methods are now used for embedding binary hash codes into distance metric functions [1, 7], thereby allowing different types of image search, such as near-duplicate search, example-based target recognition, posture estimation, and feature matching.

Kernel functions have been used widely for image feature extraction and visualization. LSH function clusters are also important for searching similar targets. However, LSH and its variant functions cannot be embedded directly to search data with many powerful kernel functions, including those designed specifically for images. Therefore, the lack of high-performance fast search algorithms is problematic for flexible kernel functions. Some other problems that exist in LSH function clusters may be neglected. However, in image search and recognition tasks, the common problems are as follows.

(A) When kernel functions are introduced into LSH, the linear relationship between the original nearest neighbor search and LSH no longer applies. The kernel functions in video images are generally nonlinear, so it is necessary to solve the problem of applying nonlinear search to large-scale image classification. Recent studies [9] indicate that the training time for a support vector machine (SVM) has a linear relationship with the sample size. These methods can be extended to large-scale data sets, online learning, and structure learning. Nonlinear SVM can be regarded as a linear SVM operating in an appropriate feature space. Hence, at least theoretically, it is feasible to extend fast algorithms to more general models. Zheng and Ip [2] proposed sparse, intersection kernel feature maps that can increase the speed by 1000 times for an SVM classifier. Thus, we propose homogeneous kernel maps for roughly estimating all additive homogeneous kernels. In addition to the intersection kernels, Hellinger, Chi2, and Jensen-Shannon kernels are included. We combine these maps with the random Fourier features described by Xiong et al. [10], and we obtain a rough estimation of the Gaussian radial basis function (RBF) variants of the additive kernels [11]. In fact, these kernel functions are particularly suitable for data in a prior probability distribution or in the form of a normalized histogram, for example, a visual vocabulary bag [12] or spatial pyramid model [13–15].

(B) A workable distance metric learning function can accurately reflect the potential relationships between similar images, such as category tags or other hidden parameters, based on a small distance metric, whereas a larger distance metric can be used for unrelated images. However, a universal LP distance is not suitable for learning problems where the data representation is already given. Specific distance functions can be learned only by using metric learning algorithms with some given information, where they can be combined with nearest neighbor search to avoid direct exact search in a high-dimensional space.

The remainder of this paper is organized as follows.

First, we propose a kernelized LSH algorithm based on the explicit mapping of additive homogeneous kernels (HKLSH). The extracted feature histograms are then used for feature mapping in an explicit additive kernel space. After transformation, the feature vectors are used as the input feature vectors for KLSH. This method resolves the speed problem with nonlinear transformation in the kernel space using KLSH, as well as reducing the unstable performance caused by the locality sensitivity of LSH, and thus the search precision is improved.

Second, we describe a method for scalable image search based on metric learning, which uses a learned Mahalanobis

distance function in approximate similarity search in order to capture the potential relationships between images. We discuss the embedding of metric learning parameters into LSH in order to guarantee the approach functions in linear time. Homogeneous feature maps are used to reduce the unstable performance caused by locality sensitivity and to increase the precision. Compared with conventional metric learning algorithms, the proposed method delivers more efficient search. Our experiments confirmed the higher precision and efficiency of the proposed method compared with other nearest neighbor search algorithms based on hash function. This method is particularly suitable for search in large databases.

2. Image Classification Model Based on Homogeneous Kernel Scalable Hashing

2.1. Analysis of Kernel Feature Maps. Kernel feature maps are usually constructed when processing low-dimensional, linear inseparable data. Thus, the data in a low-dimensional space are mapped onto a high-dimensional (Hilbert) space and denoted by $\langle \cdot, \cdot \rangle$, where $\forall a, b \in R^D : K(a, b) = \langle \psi(a), \psi(b) \rangle$ represents

$$K(a, b) = \langle \psi(a), \psi(b) \rangle \quad \forall a, b \in R^D. \quad (1)$$

Bochner's theorem (see (1)) can be used to calculate the feature maps for static kernels. In order to calculate the feature maps and obtain the feature maps for approximately homogeneous kernels, we use Bochner's theorem and extend it to a γ -homogeneous kernel. Moreover, all of the closed feature maps commonly used for homogeneous kernels can be obtained.

When the homogeneous kernel is a positive definite kernel [16], its signature is also an expression of a positive definite kernel. This lemma also applies to static kernels. By combining (1) and Bochner's theorem, the approximate feature map of $k(a, b)$ is constructed as given in

$$\begin{aligned} k(a, b) &= (ab)^{\theta/2} \int_{-\infty}^{+\infty} e^{-i\omega\lambda} \kappa(\omega) d\omega, \quad \theta = \log \frac{b}{a}, \\ &= \int_{-\infty}^{+\infty} \left(e^{-i\omega \log a} \sqrt{a^\theta \kappa(\omega)} \right)^* \left(e^{-i\omega \log b} \sqrt{b^\theta \kappa(\omega)} \right) d\omega. \end{aligned} \quad (2)$$

For most machine learning kernels, the kernel densities and approximate feature maps can be obtained in the same manner [17, 18].

Before calculating the feature maps of continuous kernels, we review Bochner's theorem and static kernels.

γ -Homogeneous Kernels. For γ -homogeneous kernel $k(x, y)$, a similar result is derived. Bochner's theorem is employed starting from (2):

$$\begin{aligned} k(x, y) &= (xy)^{\gamma/2} \int_{-\infty}^{+\infty} e^{-i\omega\lambda} K(\omega) d\omega, \quad \lambda = \log \frac{y}{x}, \\ &= \int_{-\infty}^{+\infty} \left(e^{-i\omega \log x} \sqrt{x^\gamma K(\omega)} \right)^* \\ &\quad \cdot \left(e^{-i\omega \log y} \sqrt{y^\gamma K(\omega)} \right) d\omega. \end{aligned} \quad (3)$$

We obtain the feature map:

$$\psi_w(a) = e^{-iw \log a} \sqrt{a^\gamma \kappa(\omega)}. \quad (4)$$

2.2. Feature Maps for Homogeneous Kernels. However, the feature maps described in Section 2.1 cannot be used directly because they are continuous functions where a low-dimensional or sparse approximation is required [19].

Feature maps for homogeneous/static kernels are derived from (4). The regular discrete feature maps are derived from (4). The simplified form is given and that for static kernels is as follows:

$$\widehat{\Psi}_j(x) = \begin{cases} \sqrt{\widehat{k}_0}, & j = 0, \\ \sqrt{2\widehat{k}_{(j+1)/2}} \cos\left(\frac{j+1}{2}Lx\right) & j > 0 \text{ odd}, \\ \sqrt{2\widehat{k}_{j/2}} \sin\left(\frac{j}{2}Lx\right) & j > 0 \text{ even}. \end{cases} \quad (5)$$

And that for a γ -homogeneous kernel is as follows:

$$\widehat{\Psi}_j(x) = \begin{cases} \sqrt{x^\gamma \widehat{k}_0}, & j = 0, \\ \sqrt{2x^\gamma \widehat{k}_{(j+1)/2}} \cos\left(\frac{j+1}{2}L \log x\right) & j > 0 \text{ odd}, \\ \sqrt{2x^\gamma \widehat{k}_{j/2}} \sin\left(\frac{j}{2}L \log x\right) & j > 0 \text{ even}. \end{cases} \quad (6)$$

2.3. KLSH Based on Homogeneous Kernel Maps. KLSH is used to construct data connections. The basic principle of KLSH was introduced above [20]. In this section, we combine explicit homogeneous kernel maps with KLSH to establish the cost algorithm, as described in the following. Similar to LSH, the establishment of the hash function is also a major concern in KLSH. Thus, in order to calculate the collision probability between the input query items and database items, we must calculate the similarity between any two items in the database [20, 21].

According to previous studies [22, 23], the definition of the LSH function can be extended from (6) into the following form:

$$h_{\vec{r}}(m) = \begin{cases} 1, & \text{if } \vec{r}^T m \geq 0 \\ 0, & \text{else.} \end{cases} \quad (7)$$

First, \vec{r} is constructed from a database subset. According to the central limit theorem, for some subset items chosen from the entire database S , the kernel data samples must have mean μ and variance Σ , which obey the normal distribution. Thus, the variable z_a can be written as follows.

$$z_a = \frac{1}{t} \sum_{i \in S} \varphi(\widehat{\Psi}(m_i)). \quad (8)$$

As the variable a increases, the vector $\vec{z}_a = \sqrt{t}(z_a - \mu)$ must obey the normal Gaussian distribution \vec{r} , which is obtained by the following whitening transform:

$$\vec{r} = \Sigma^{-1/2} \vec{z}_a. \quad (9)$$

TABLE 1: HKLSH algorithm.

(1) Process data
(a) Obtain matrix K from the database throughout the n points
(b) Obtain e_s by randomly sampling a subset from $\{1, 2, \dots, n\}$
(c) Project onto the t th subset to obtain $h_t(\varphi(\widehat{\Psi}(m)))$
(d) Obtain $w = K^{1/2} \cdot e_s/t$
(e) Project $w(i)$ onto the homogeneous kernel space
(f) Obtain the hash bucket $g_j(m)$
(2) Query processing
(a) Obtain the same hash bucket from the database
(b) Use ANN (approximate nearest neighborhood) search for query matching

The LSH function is obtained as

$$h(\varphi(\widehat{\Psi}(m))) = \begin{cases} 1, & \text{if } \varphi(\widehat{\Psi}(m))^T \Sigma^{-1/2} \vec{z}_a \geq 0 \\ 0, & \text{else,} \end{cases} \quad (10)$$

$$h(\varphi(\widehat{\Psi}(m))) = \sum_{i=1}^n w(i) \left(\varphi(\widehat{\Psi}(m_i))^T \varphi(\widehat{\Psi}(m)) \right), \quad (11)$$

where $w(i) = \sum_{j=1}^n K_{ij}^{-1/2} \varphi(\widehat{\Psi}(m_j))^T \vec{z}_a$.

Using the formula derived for $w(i)$, the random hyperplane vector $\vec{r} = \sum_{i=1}^n w(i) \varphi(\widehat{\Psi}(m_i))$ is obtained and it obeys the normal Gaussian distribution. After substituting (11) into $w(i)$, we obtain

$$w(i) = \frac{1}{t} \sum_{j=1}^n \sum_{l \in S} K_{ij}^{-1/2} K_{jl}. \quad (12)$$

The coefficient \sqrt{t} is omitted to obtain the simplified expression as (12), where e_s is the unit vector in the database set S . Thus, (13) is obtained for the hash function of the input kernel:

$$w = K^{1/2} \cdot \frac{e_s}{k}, \quad (13)$$

$$h(\varphi(\widehat{\Psi}(m))) = \text{sign} \left(\sum_{i=1}^n w(i) \kappa(\widehat{\Psi}(m), \widehat{\Psi}(m_i)) \right), \quad (14)$$

where κ is the kernel map matrix of $\widehat{\Psi}(m)$ and $\widehat{\Psi}(m_i)$ in the space. After several iterations, we obtain the hash bucket.

Several iterations are performed for query matching in order to obtain the optimal parameters. Table 1 shows the algorithm known as LSH based on homogeneous kernel maps (HKLSH), where

$$g_j(m) = [h_1(\varphi(\widehat{\Psi}(m))), h_{2,j}(\varphi(\widehat{\Psi}(m))), \dots, h_{l,j}(\varphi(\widehat{\Psi}(m))), \dots, h_{t,j}(\varphi(\widehat{\Psi}(m)))] \quad (15)$$

$$1 < l < t, 1 < j < b.$$

2.4. Image Search Model Based on a Homogeneous Kernel Map with Metric Learning. For a given training sample set, a parameterized Mahalanobis distance function can be learned with specific tags or pairwise constraints. The learned information is embedded and coded into the random hash function. Thus, when using metric learning, the collision probability in a hash table will be greater when the similarity of the inputs is higher.

2.4.1. Similarity Search Based on Semisupervised Hashing. Similar to LSH, semisupervised hashing is a locality-sensitive hash table search method. When integrated with the K -nearest neighbor method, semisupervised hashing can greatly increase the speed of image classification and search. The basic principles of LSH were described in Section 2.3, so no further details are provided. In this section, we combine information theoretic metric learning (ITML) [1] with LSH.

As described in a previous study [1], ITLM is either linear and explicit or nonlinear and implicit. We focus on the former type of ITLM.

Next, we consider how LSH accepts the matrix learned using ITML, that is, how to utilize the constraint conditions obtained by LSH. For the learned matrix A , we have $A = G^T G$ and the hash function is expressed as

$$h[\varphi(x)] = \begin{cases} 1, & r^T G \varphi(x) \geq 0 \\ 0, & \text{else,} \end{cases} \quad (16)$$

where the vector r is randomly selected from those that obey a d -dimensional Gaussian distribution with zero mean and unit variance. This selection can be restricted by using G . The hash function itself carries the information obtained from metric learning, which is the constraint condition from the last step.

The hash function parameterized by G is used to derive the following similarity relationship:

$$\begin{aligned} & \Pr [h_{r,A}(\varphi(x_i)) = h_{r,A}(\varphi(x_j))] \\ &= 1 - \frac{1}{\pi} \cos^{-1} \left(\frac{\varphi(x_i)^T A \varphi(x_j)}{\sqrt{|G\varphi(x_i)| |G\varphi(x_j)|}} \right). \end{aligned} \quad (17)$$

This algorithm can satisfy the requirement for LSH given in (17) under Mahalanobis metric learning, where A is calculated as described previously [1, 3].

3. Experimental Results and Analysis

3.1. Image Databases. The Caltech-101 and Photo Tourism data sets were used in our experiments. In the first experiment with KLSH based on homogeneous kernel maps, we used image patches ($n = 100000$) from the Notre Dame Cathedral class in Photo Tourism.

Further details of the experiments using the same data sets are not repeated. We also performed experiments with two other databases: Flickr and its subsets and Tiny-Image.

Flickr. The second similarity search experiment was based on semisupervised hashing. One of the data sets comprised 5400 photos related to 18 tourist attractions obtained from Flickr. Similar to the first experiment, three European cities were selected, namely, Rome, London, and Paris, each with one tourist attraction.

Tiny-Image. The data set comprised 79,302,017 images of commercial products, which were crawled using the Google search engine. We used 1.7 million photos from Tiny-Image as the training data set, where each photo was in $32 * 32$ pixel format. This experiment involved content-based image retrieval.

3.2. Experimental Setup. The experiment comprised two processes, as follows.

(1) Hashing Based on Homogeneous Kernel Maps. Multiscale SIFT feature extraction was performed first to extract local patches from Caltech-101. Four scales were considered with four pixels for each scale. K -means clustering was used for vector quantization based on 300 visual words and the BOF packet was obtained. Feature histograms were calculated to obtain visual word histograms in the 2100-dimensional subdomain for each image ($1 * 1$, $2 * 2$, and $4 * 4$), which were used to replace the original representation of SIFT features. In the second step, the χ^2 kernel was taken as the feature map function of the γ -homogeneous kernels ($\gamma = 1/2$). Explicit feature maps were calculated for the homogeneous kernels based on the feature histograms obtained in the first step. In the third step, an RBF kernel function based on the L2 function was used as the distance-based kernel function to further process the explicit feature maps obtained in the second step. The result obtained was used as the input for KLSH. The same parameter configurations employed in the two experiments above were used for the Tiny-Image data set. The Tiny-Image data set was very large, so a large value of ε was applied. In order to compensate for the loss of precision due to the large value of ε , the number of hash buckets was increased in order to retrieve more similarity matches. For each image, $b = 300$ and 1000 images were randomly selected as the query items. The parameters were kept constant at $t = 30$ and $n = 300$ for the standard KLSH algorithm. GIST features were used as the inputs for both HKLSH and standard KLSH [24].

(2) Similarity Search Based on Semisupervised Hashing. GB features were first calculated for the data set [25], before employing K -means clustering to obtain the visual word packet and histograms. Next, the feature maps for additive homogeneous kernels (γ -homogeneous kernel χ^2 with $\gamma = 1/2$) were calculated in the feature space to overcome the instability caused by locality sensitivity, as well as increase the precision and speed. ITML was implemented to obtain the learning matrix, which was subjected to operations as well as the eigenmatrix, where the result was used as the input for LSH. Finally, the K -nearest neighbor algorithm was implemented and the accuracy was estimated. The proposed algorithm is referred to as the hom+ml hashing algorithm.

Several parameters were varied in order to assess the classification and retrieval performance of each algorithm based on different data sets: (1) number of sample points n , number of hash tables b , and number of hash functions in each hash table t ; (2) selection of the feature map function for γ -homogeneous kernels; (3) parameter ε , which affects the retrieval speed and accuracy; (4) number of classes K in the K -nearest neighbor algorithm; and (5) number of training examples in each class of the data set.

The proposed algorithm was tested in visual classification and similarity search tasks. The experimental platform comprised an Intel® Core™ i7-4710MQ Processor (3.2 GHz frequency), with 16 GB of memory and a 2 TB hard drive.

4. Results and Analysis

There is an increasing need to retrieve entire images or local patches from large databases. In the first experiment, we verified the performance of HKLSH using several data sets. The proposed HKLSH method allows the embedding of unknown features and it also guarantees that the proposed approach runs in linear time. Our method then utilizes these features to conduct vision search tasks to locate the most similar items in a database. This method is more effective and accurate than linear scan search.

Semisupervised hashing (hom+ml hashing) was applied to the Caltech-101 and Flickr data sets. The results of the experiments showed that, compared with other methods, the accuracy was improved greatly and the speed was considerably faster than that of ordinary LSH algorithms.

4.1. HKLSH

4.1.1. Caltech-101 Data Set. In the previous section, we described the experiment conducted based on the Caltech-101 data set, and thus the details are not repeated. Many important studies have analyzed image representation in terms of kernels [25], where different kernels have been employed for image feature extraction from various data sets [24, 26–28]. Grauman and Darrell [24] proposed image matching based on a pyramid match kernel for feature histograms. Berg et al. [26] described a method that uses the CORR (corresponding matching for geometric blur) kernel of image local features based on fuzzy geometry as a distance-based match kernel to compute the similarity. Figure 1 shows the experimental results obtained based on the Caltech-101 data set where 15 images from each class were used as the training samples.

Figure 1 shows that the precision was directly proportional to different parameters in most cases. In general, the accuracy was stable and the precision was strongly associated with increases in n but weakly associated with t , where there were some fluctuations. From Figure 1, we can see that the performance value of SIFT-RBF&Chi2 Lin Scan is straight line, as it would not be influenced by the parameters n , b , and t . Meanwhile, in Figure 1(c), we can see that the performance has twist after $t > 30$, which means the accuracy will decrease after parameter $t > 30$ and if the value of t (hash function value) is too big, the performance will decline. According to

Figure 1, the best search performance was obtained using a combination of $n = 300$, $b = 300$, and $t = 30$.

The parameter ε is used to balance the search accuracy and speed, and it had a direct impact on the search results. As shown in Figure 2, when $\varepsilon = 0.2$ (only 21.86% of the data were searched; the value could be computed by formula $M = 2n^{1/(1+\varepsilon)}$), the accuracy was about 68% and the accuracy of the linear scan search was 71%, which was higher than the precision of CORR-KLSH [20] (58.5%). Overall, an appropriate combination of n , b , and t led to highly stable accuracy.

In terms of the search speed, the percentage of items searched in the database remained constant with the changes in t and n ; that is, 21.86% of the items were searched in the database using HKLSH when $\varepsilon = 0.2$, 4.8% of the items were searched in the database using HKLSH when $\varepsilon = 0.5$, and only 0.67% of the items were searched when $\varepsilon = 1.5$. When it was applied to larger data sets, the proposed method searched an even lower percentage of the items.

The comparisons using different parameters indicated that various combinations of parameters directly affected the classification accuracy for a large data set.

Figure 3 compares the performance of the proposed method with the test data set and the results obtained in previous studies [24, 29–32]. Table 2 shows that the accuracy obtained using the proposed method with $T = 15$ and $T = 30$ (i.e., the number of training samples for each class was 15 and 30, respectively) was lower than that reported in some previous studies [33, 34], but higher than the accuracy rates of 61% [1] ($T = 15$) and 69.6% in other studies. Compared with studies conducted many years ago, the accuracy improved by up to 16%.

In order to determine the optimal parameters for the NN search, we attempted to balance the CPU time and performance. Curves for the CPU time versus search performance were plotted for different values of K ($K = 2, 3, \dots, 20$). Figures 4 and 5 show the changes in the search accuracy with the CPU time for different values of K .

Figure 4 shows that the accuracy decreased as the value of K increased. Figure 5 shows that the CPU time increased as the value of K increased. Thus, we set $K = 2$ in order to balance the CPU time and search performance.

In a previous study [35], KPCA and LSH were combined in order to conduct hashing based on KPCA. This method has apparent advantages; however, in this algorithm, the input information is lost during dimensionality reduction for the features when using KPCA. By contrast, our proposed HKLSH method can maintain the integrity of the input information when implementing LSH; thus, our method is more accurate than KPCA combined with LSH [35].

Table 3 shows the accuracy and time obtained using the χ^2 kernel and other kernels with MB maps, homogeneous kernel maps, KPCA maps, and libsvm maps (averages based on five replicates). The accuracy of the linear kernel (slightly above 40%) was the lowest and it was 15–20% lower than the accuracy of the other kernels. The accuracy was highest for the χ^2 and JS kernels, followed by the Hellinger and intersection kernels. The accuracy of the latter two kernels was only slightly lower (by 1–2%) compared with that of the

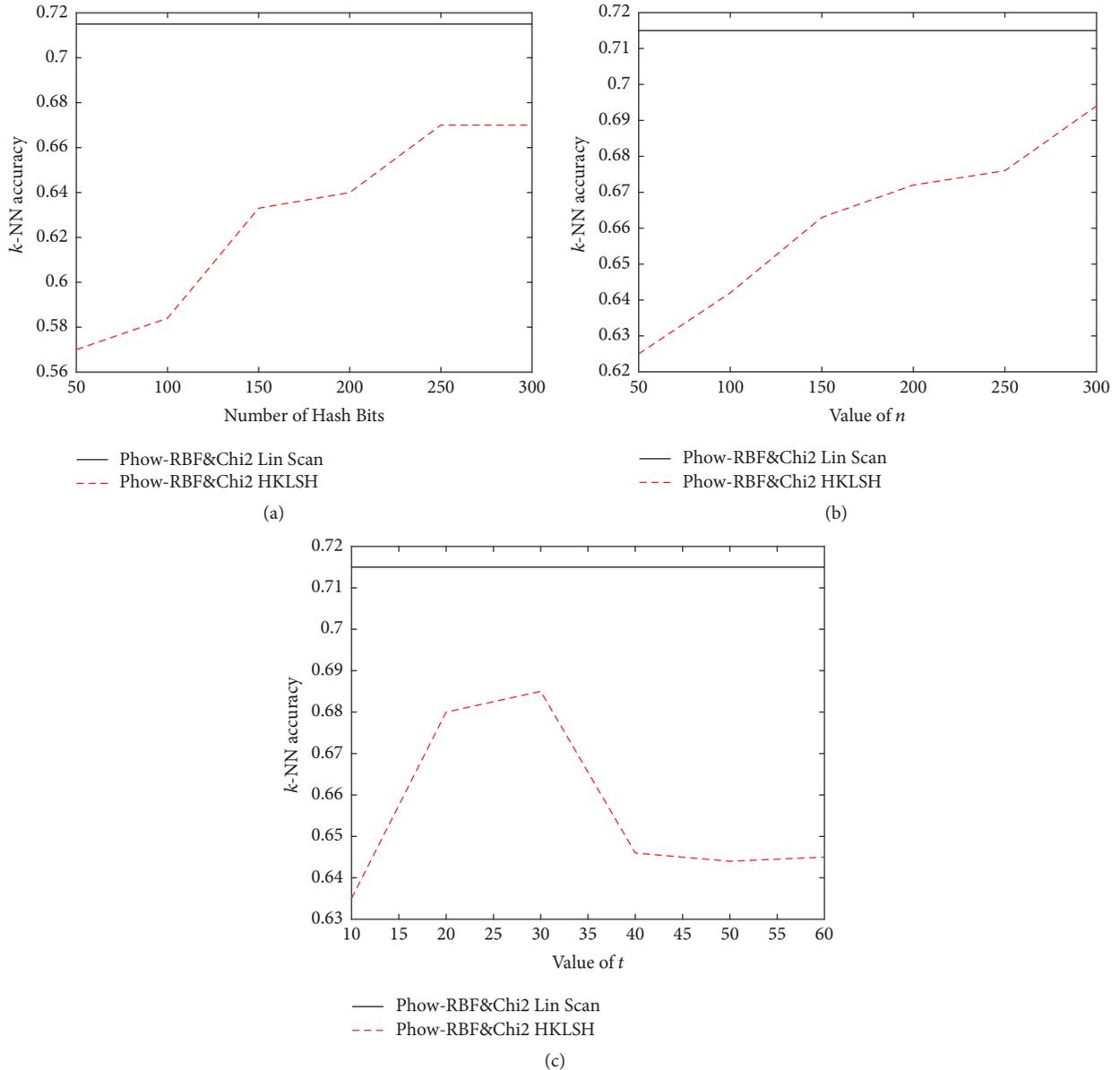


FIGURE 1: Performance comparison using the Phow-RBF- χ^2 kernel based on a γ -homogeneous kernel map with different values for b , n , and t .

TABLE 2: Performance comparison based on the Caltech-101 data set.

Number of training samples	Proposed method	[25]	[29]	[30]	[24]	[31]	[32]
15	68.5 ± 1.2	59.1	56.4	52	51	49.5	44
30	73.2 ± 0.8	66.2	64.6	N/A	56	58.2	63

former two kernels. Our method and KPCA approximation were superior to the libsvm kernel. The homogeneous kernel with $\gamma = 1/2$ was slightly better than that with $\gamma = 1$, probably because the former can reduce the effect of a large peak on the feature histograms. As expected, the K -nearest neighbor-based hashing algorithm with approximation using homogeneous kernels was much faster than the libsvm classifier using a single kernel, where the search speed increased in a

linear manner as the size of the data set increased. Compared with linear brute force search, the speed of our method was improved by over 10 times. This advantage would be more apparent if our method is applied to larger data sets.

4.1.2. Local-Patch Index Database. The aim of the task was to extract all relevant local patches, so the performance was assessed based on the accuracy and recall. The distance metric

TABLE 3: Performance comparison based on homogeneous kernel maps for each kernel function in terms of the training time (s) and accuracy (%).

	χ^2		JS		Intersection		Hellinger	
	Accuracy	time	Accuracy	time	Accuracy	time	Accuracy	time
MB	62.5	220	62.15	228	61.6	254	65.6	205
KPCA	64.5	679	65.1	1460	62.74	752	64.5	280
libsvm	64.3	1367	64.6	6783	62.4	1326	63.87	690
Hom-kernel	67.44	312	67.32	446	65.7	385	65.4	159
1/2-hom-kernel	73.25	375	72.92	486	71.6	487	71.2	184

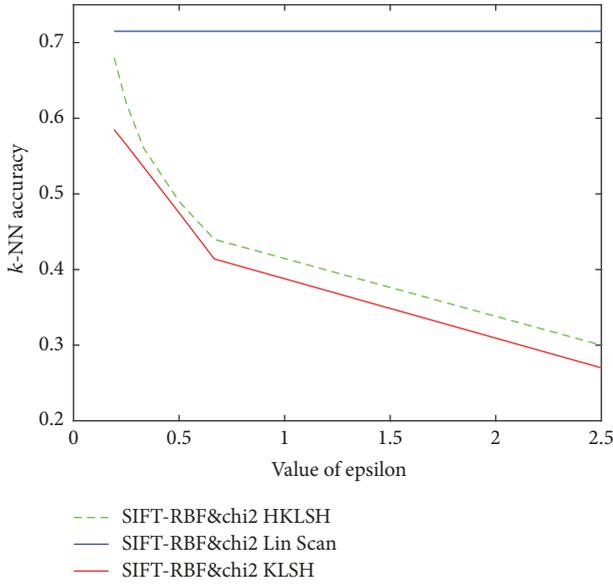


FIGURE 2: Comparison of classification accuracy corresponding to different values of ϵ .

functions used comprised the Euclidean distance (L2) metric function and the Gaussian RBF kernel based on the χ^2 homogeneous kernel (the parameter $\gamma = 0.5$), where the standard LSH was suitable for the former and the proposed HKLSH was employed for the latter considering the difficulty of computing the potential embedded information for the data and the infinite dimensionality of the data.

Figure 6 compares the accuracy and recall performance obtained using different numbers of retrieved images. In Figure 6(a), the two almost overlapping curves in the lower position correspond to the recall obtained using linear scan search based on the Euclidean distance and LSH based on the Euclidean distance, respectively. The two higher curves correspond to the recall obtained using linear scan search based on the χ^2 -RBF kernel and HKLSH based on the χ^2 -RBF kernel, respectively. HKLSH increased the accuracy by using more powerful distance-based kernels to extract the image features. When $\epsilon = 1.5$, HKLSH only needed to search 0.12% of the items in the database on average, which represents a major improvement compared with the linear scan search method. In Figure 6(b), the two lower curves correspond to the accuracy of linear scan search based on the Euclidean

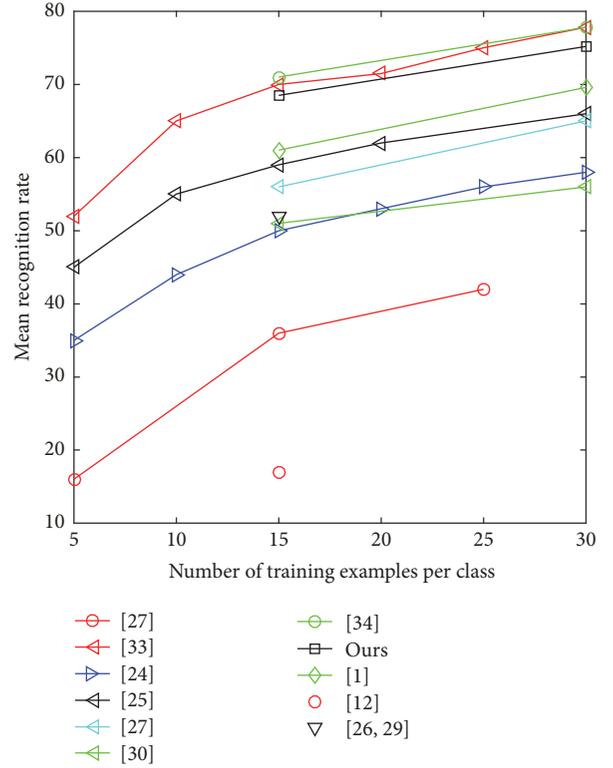


FIGURE 3: Performance comparison based on the Caltech-101 data set using different numbers of training images.

distance and that of LSH based on the Euclidean distance, respectively, whereas the two higher curves correspond to the accuracy of linear scan search based on the χ^2 -RBF kernel and that of HKLSH based on the χ^2 -RBF kernel, respectively. According to Figures 6(a) and 6(b), we can see that the recall obtained using the hash function was almost equal to that with linear scan search. Moreover, HKLSH based on the χ^2 -RBF kernel achieved similar accuracy to that with linear scan search based on the χ^2 -RBF kernel in sublinear time, which represents a major improvement compared with existing algorithms.

The time to search for matches with each query item was about 2.2×10^{-2} s, which represented a decrease of about 62.8×10^{-2} s compared with KLSH (6.5×10^{-1}). This process required 5.8 s using the linear scan search.

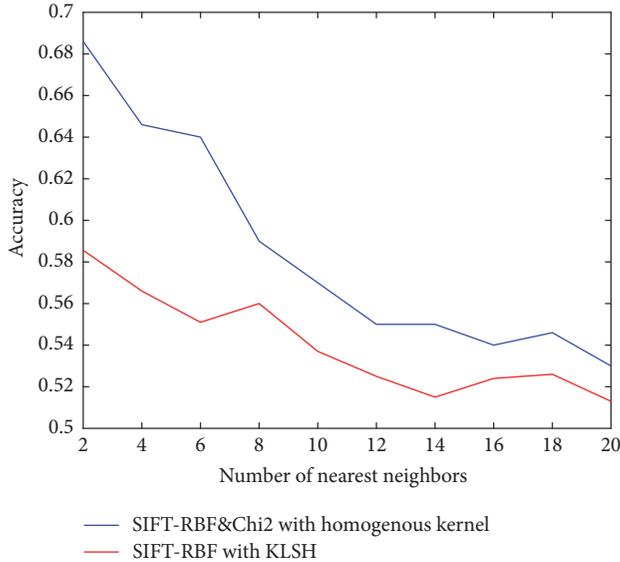


FIGURE 4: Comparison of classification accuracy corresponding to different values for K .

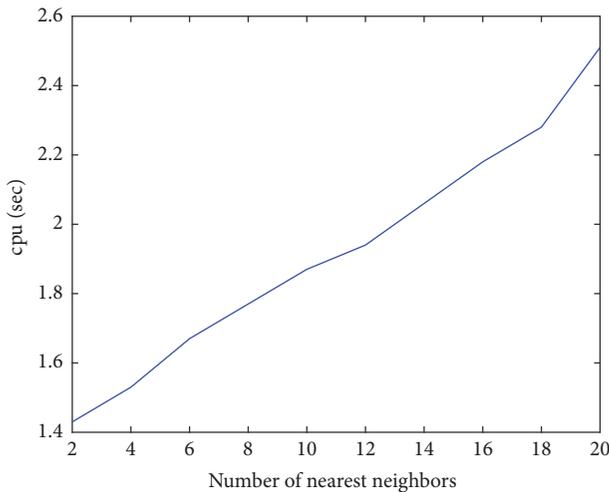


FIGURE 5: CPU workload versus different values for K .

4.1.3. Tiny-Image Data Set. Experiments were also conducted using a subset of the Tiny-Image data set in order to verify the scalability of our method to large data sets.

In the experiments based on the Caltech-101 and Local-Patch data sets, image search was performed using binary hash codes in order to compare different hash algorithms. For the Tiny-Image data set, layered search was implemented to yield similar images.

The retrieved images were ranked according to previously defined criteria [4] given the query image q and similarity metric function. The top k images retrieved for the query image q were assessed based on the proportion of good neighbors (proportion of the most qualified images). Our method was compared with the standard KLSH in terms of the search speed and the result confirmed the high performance of our method with large data sets.

We used the same set of parameters employed in the experiment described above, but a larger value was selected for ϵ because of the large scale of the data set. The number of hash buckets was increased to compensate for the loss of precision caused by a larger value of ϵ and to obtain more matches.

Figures 7(a) and 7(b) show the proportions of good neighbors (defined as retrieved images with the same tags as the query image) obtained using different numbers of hash bits and the number of images retrieved, which show that the proposed HKLSH was superior to the standard KLSH in terms of both indexes. Figure 7(b) shows curves for the numbers of images retrieved with different numbers of hash bits. HKLSH only needed to search 0.56% of the items in the database in order to achieve good precision, where the time needed to search for matches with each query item was about 3.2×10^{-1} s, which represented a decrease of about 1.3×10^{-1} s compared with KLSH (4.5×10^{-1}). This process required 47.5 s using the linear scan search.

4.2. Match Results Based on Semisupervised Hashing

4.2.1. Caltech-101 Data Set. As shown by the performance comparison in Table 4, the precision was about 72.5% based on the Caltech-101 data set using our method ($T = 30$), which was lower than the precision (78%) reported in previous studies [33, 34]. However, the precision of our method was higher compared with other algorithms. The classification error rate and search efficiency using different numbers of hash bits were also determined for different algorithms.

Figure 8(a) compares the classification error rate using different numbers of hash bits with linear scan search based on the L2 Euclidean distance, LSH based on the L2 Euclidean distance, and K -nearest neighbor classification using our method. Figure 8(b) shows the proportion of images retrieved from the database using the LSH based on the L2 Euclidean distance and our method. These results show that our method performed better than the ordinary LSH in terms of both the classification error rate and search efficiency. As shown in Figure 8(b), the proportion of images retrieved from the database using our method decreased by about 40% when using the same number of hash bits compared with ordinary LSH, which represents a considerable improvement in the classification efficiency using our method.

4.2.2. Flickr Data Set. Figure 9(a) shows the functional relationship between the classification error rate and the parameter ϵ with different algorithms. According to Figure 9(a), as ϵ increased, the classification error rate with our method was far lower than that obtained using LSH based on the L2 Euclidean distance and linear scan search based on the L2 Euclidean distance, but it was similar to the classification error rate with Hom-ML hashing.

Figure 9(b) shows the functional relationship between the proportion of items searched in the database and the parameter ϵ using two algorithms. Apparently, ϵ controlled the time needed for retrieval. These results show that a lower proportion of items were searched in the database using our method than LSH based on the L2 Euclidean distance with

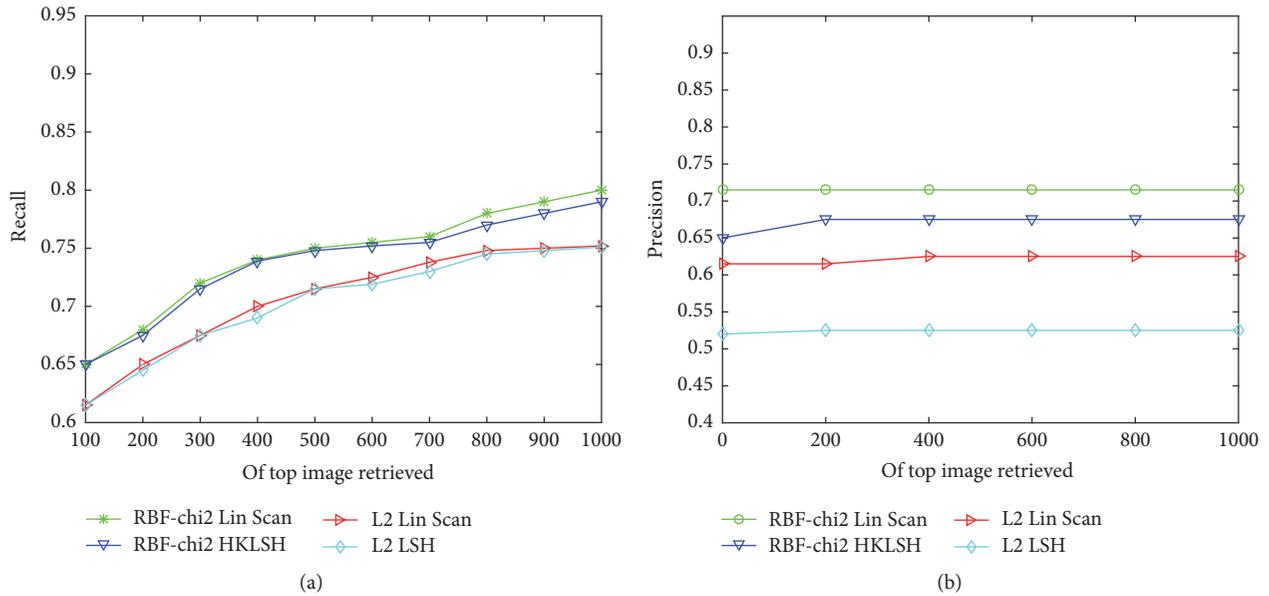


FIGURE 6: (a) Comparison of the recall obtained using different numbers of retrieved images based on the Local-Patch database. (b) Comparison of the accuracy obtained using different numbers of hash bits based on the Local-Patch database.

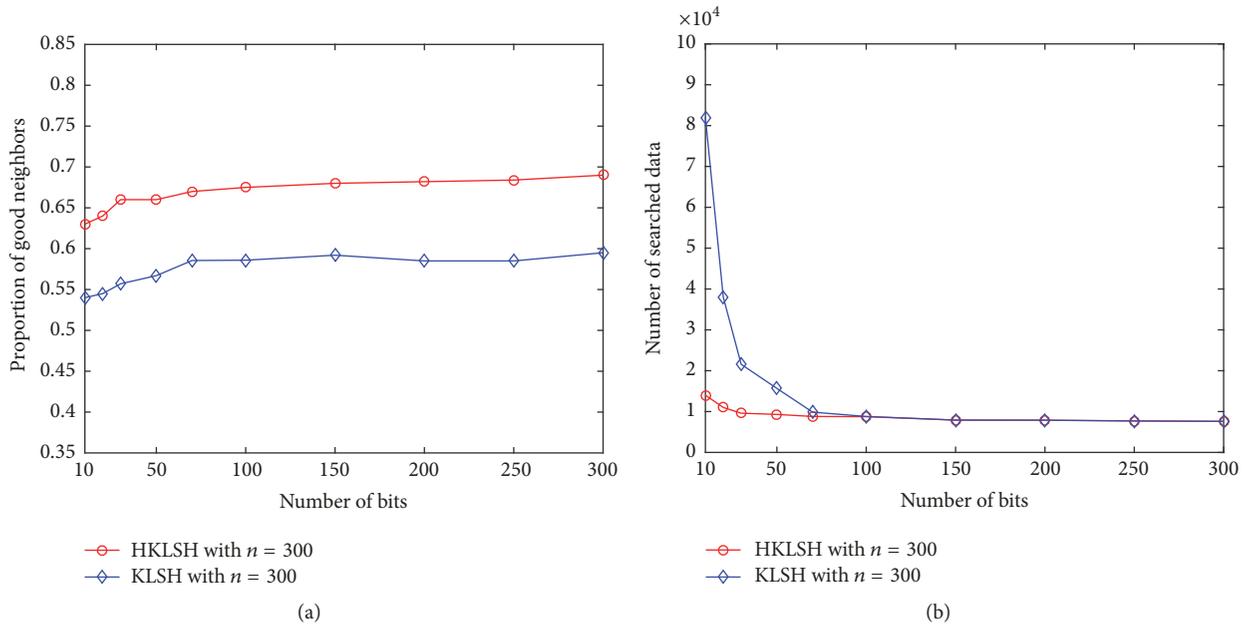


FIGURE 7: (a) Proportion of good neighbors obtained using different numbers of hash bits based on the Tiny-Image database. (b) Numbers of images retrieved using different number of hash bits based on the Tiny-Image database.

TABLE 4: Performance comparison based on the Caltech-101 data set.

Number of training samples	Our method	[25]	[29]	[31]	[32]	[33]	[34]
15	69.5 ± 0.25	62.5	61.5	NA	NA	71.5	72.5
30	72.5 ± 0.75	66.2	64.6	58.2	63	77.8	78

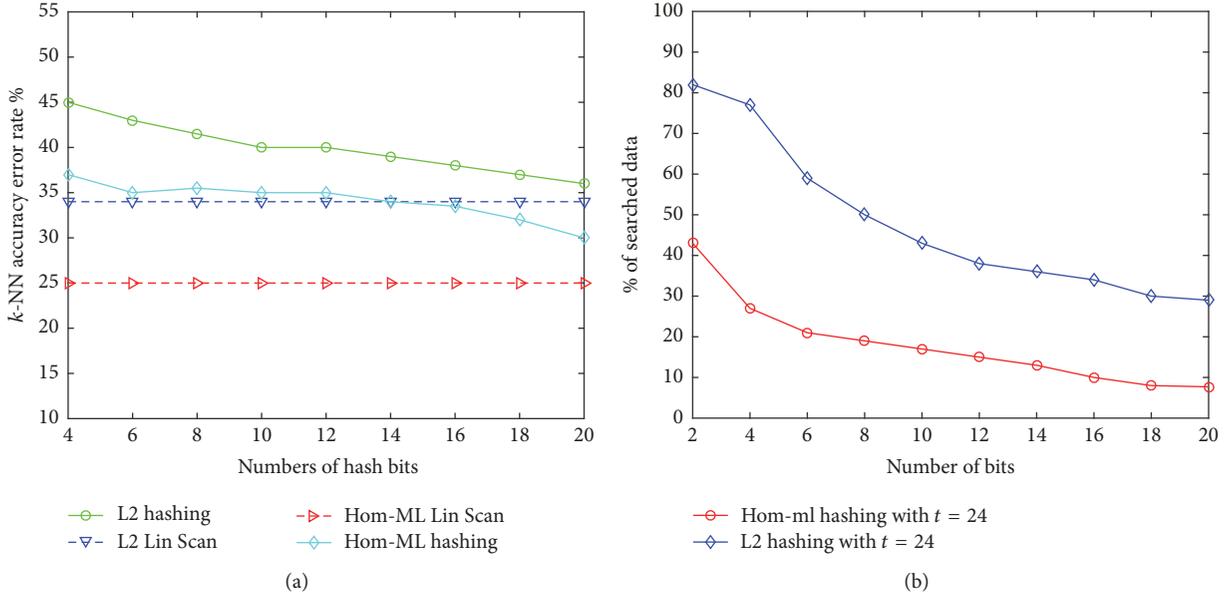


FIGURE 8: (a) Classification error rate using different numbers of hash bits based on the Caltech-101 database. (b) Proportion of good neighbors retrieved using different numbers of hash bits based on the Caltech-101 database.

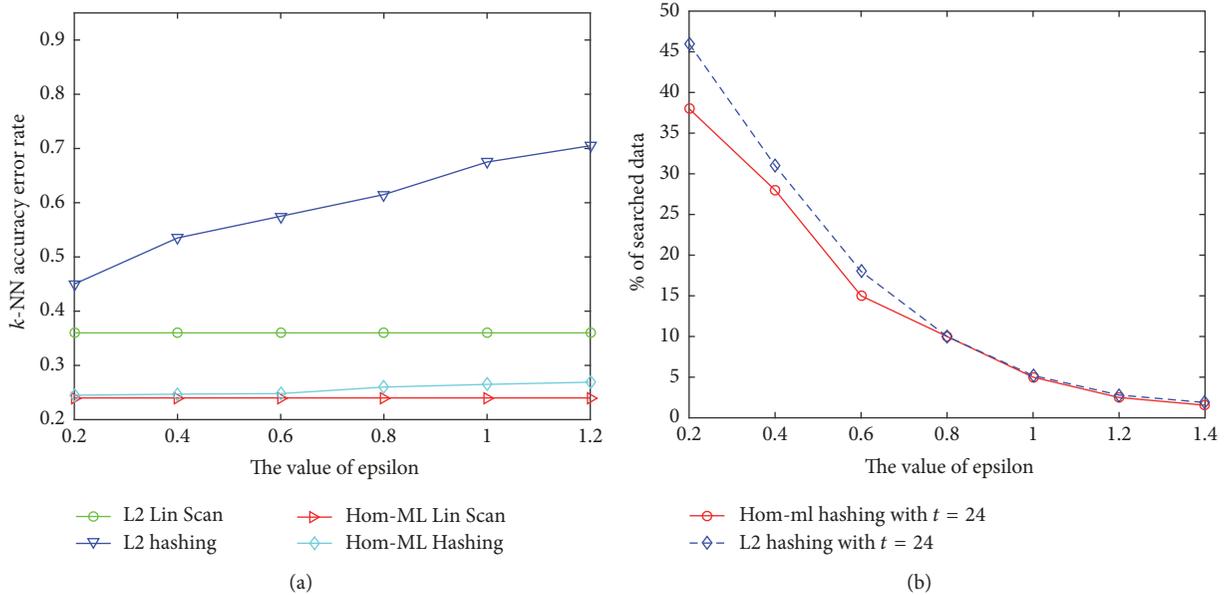


FIGURE 9: (a) Classification error with each algorithm using different values of ϵ based on the Flickr data set. (b) Proportion of good neighbors retrieved using each algorithm with different values of ϵ based on the Flickr data set.

the same value of ϵ , and the precision was much higher. Thus, when only a small proportion of items were searched in the database, our method achieved higher accuracy. Our method only required 1.7 s to search for matches with each query item, whereas the ordinary LSH required 1.95 s, which represents a reduction of about 12.8%.

5. Conclusion

In this study, we introduced homogeneous kernel maps for the approximation of kernels, including χ^2 , JS, Hellinger,

and intersection kernels, which are widely used in machine learning. The core idea is to perform kernel mapping for the additive explicit kernels and the feature histograms, where the kernel maps are used as the input for KLSH. In this method, any functions can be used as the input hash functions for LSH. The search precision of our method is not as high as that of global linear scan search, but it can greatly improve the search efficiency without sacrificing much of the precision. Moreover, there is no need to consider the distribution of the input data in our method. These features make our method suitable for searching large databases such as Flickr and

Tiny-Image. We proposed a new algorithm that integrates additive homogeneous kernel maps and explicit parameterized LSH based on the Mahalanobis distance. We tested the proposed method based on image data sets and a feature index set. Compared with existing methods, the performance of our method is superior in terms of the precision and search speed. Our experiments also demonstrated that the proposed method performed better than the standard KLSH in terms of the accuracy and operating time.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] L. Chen, Y. Gao, and X. Li, "Efficient metric indexing for similarity search and similarity joins," *IEEE Transactions on Knowledge Data Engineering*, pp. 591–602, 2015.
- [2] H. Zheng and H. H. S. Ip, "Image classification and annotation based on robust regularized coding," *Signal, Image and Video Processing*, vol. 10, no. 1, pp. 55–64, 2016.
- [3] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, "Depth-based hand pose estimation: Data, methods, and challenges," in *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15)*, pp. 1868–1876, December 2015.
- [4] G. Shakhna Rovich, T. Darrell, and P. Indyk, "Nearest-neighbor methods in learning and vision," *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 377–377, 2008.
- [5] L.-C. Hsieh, G.-L. Wu, W.-Y. Lee, and W. Hsu, "Two-stage sparse graph construction using MinHash on MapReduce," in *Proceedings of the 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '12)*, pp. 1013–1016, March 2012.
- [6] M. E. Houle and M. Nett, "Rank-based similarity search: reducing the dimensional dependence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 136–150, 2015.
- [7] C. Li, Q. Liu, J. Liu, and H. Lu, "Ordinal distance metric learning for image ranking," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1551–1559, 2015.
- [8] H. Q. Minh, L. Bazzani, and V. Murino, "A unifying framework in vector-valued reproducing kernel Hilbert spaces for manifold regularization and co-regularized multi-view learning," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 769–840, 2016.
- [9] T. Yang, M. Mahdavi, R. Jin, and S. Zhu, "An efficient primal dual prox method for non-smooth optimization," *Machine Learning*, vol. 98, no. 3, pp. 369–406, 2015.
- [10] Y. Xiong, Y. Zhu, and P. S. Yu, "Top-k similarity join in heterogeneous information networks," *IEEE Transactions on Knowledge Data Engineering*, vol. 27, no. 6, p. 1, 2015.
- [11] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, 2008.
- [12] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '04)*, pp. 59–70, July 2004.
- [13] X.-S. Wei, B.-B. Gao, and J. Wu, "Deep Spatial Pyramid Ensemble for Cultural Event Recognition," in *Proceedings of the 15th IEEE International Conference on Computer Vision Workshops (ICCVW '15)*, pp. 280–286, December 2015.
- [14] D. Li, N. Li, J. Wang, and T. Zhu, "Pornographic images recognition based on spatial pyramid partition and multi-instance ensemble learning," *Knowledge-Based Systems*, vol. 84, pp. 214–223, 2015.
- [15] E. Mavridaki and V. Mezaris, "No-reference blur assessment in natural images using Fourier transform and spatial pyramids," in *Proceedings of the IEEE International Conference on Image Processing*, pp. 566–570, 2015.
- [16] J. Kivinen, J. A. Smola, and R. C. Williamson, *Learning with kernels*, MIT Press, 2002.
- [17] J.-Y. Li and J.-H. Li, "Fast image search with locality-sensitive hashing and homogeneous kernels map," *Scientific World Journal*, vol. 2015, Article ID 350676, 2015.
- [18] A. Barla, F. Odone, and A. Verri, "Histogram intersection kernel for image classification," in *Proceedings of the International Conference on Image Processing (ICIP '03)*, pp. 513–516, September 2003.
- [19] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Advances in Neural Information Processing Systems*, vol. 20, pp. 1177–1184, 2007.
- [20] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," in *Proceedings of the Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 2130–2137, Kyoto, Japan, October 2009.
- [21] T.-Y. Lin, C.-C. Hsu, W. Y. Wang, and Y.-T. Wang, "Enhanced simultaneous localization and map building," in *Proceedings of the 2014 International Automatic Control Conference (CACIS '14)*, pp. 122–127, November 2014.
- [22] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the Association for Computing Machinery*, vol. 42, no. 6, pp. 1115–1145, 1995.
- [23] A. Chakrabarti, V. Satuluri, A. Srivathsan, and S. Parthasarathy, "A Bayesian perspective on locality sensitive hashing with extensions for kernel methods," *ACM Transactions on Knowledge Discovery from Data*, vol. 10, no. 2, article no. 19, 2015.
- [24] K. Grauman and T. Darrell, "Discriminative classification with sets of image features," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 2, pp. 1458–1465, October 2005.
- [25] S. S. Bucak, R. Jin, and A. K. Jain, "Multiple kernel learning for visual object recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1354–1369, 2014.
- [26] A. C. Berg, T. L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 26–33, June 2005.
- [27] A. D. Holub, M. Welling, and P. Perona, "Combining generative models and fisher kernels for object recognition," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, pp. 136–143, October 2005.
- [28] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, pp. 1–8, October 2007.
- [29] A. C. Berg, *Shape Matching and Object Recognition [Ph.D. thesis]*, University of California, Berkeley, CA, USA, 2005.

- [30] J. Mutch and D. G. Lowe, "Multiclass object recognition with sparse, localized features," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 11–18, June 2006.
- [31] G. Wang, Y. Zhang, and L. Fei-Fei, "Using dependent regions for object categorization in a generative framework," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 2, pp. 1597–1604, June 2006.
- [32] I. Murtza, D. Abdullah, A. Khan, M. Arif, and S. M. Mirza, "Cortex-inspired multilayer hierarchy based object detection system using PHOG descriptors and ensemble classification," *Visual Computer*, vol. 33, no. 4, pp. 1–14, 2015.
- [33] P. Gehler and S. Nowozin, "On feature combination for multi-class object classification," in *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision (ICCV '09)*, pp. 221–228, September 2009.
- [34] J. Young, "Spatial pyramid match kernels for brain image classification," in *Proceedings of the 6th International Workshop on Pattern Recognition in Neuroimaging (PRNI '16)*, pp. 1–4, June 2016.
- [35] J. Yang, X. Gao, and D. Zhang, "Kernel ICA: an alternative formulation and its application to face recognition," *Pattern Recognition*, vol. 38, no. 10, pp. 1784–1787, 2005.

